T

Tunlease

Debug webhooks locally through a fixed callback URL that stays the same while requests are routed to localhost.

About

Tunlease is a webhook debugging tool for local development that lets you use a real, unchangeable callback URL without redeploying or creating a new endpoint. It claims a path on an existing callback URL, then routes that traffic to localhost while the rest of the app can keep serving normally. That makes it practical for staging webhook flows, local callback testing, and fast iteration when providers expect a fixed URL.

What is Tunlease

Tunlease is a webhook debugging tool built to expose a local service behind a stable

changes, you claim one path on an existing endpoint and point it at `localhost`. The app keeps the callback URL fixed, which is the core reason people use a webhook debugging tool like Tunlease during development.

The site presents Tunlease as self-hostable, open source, and MIT licensed. It is aimed at staging webhooks and local callback testing rather than general-purpose tunneling. The workflow is simple: provider traffic reaches Tunlease, Tunlease forwards the chosen path to your local server, and unrelated paths continue to serve the real application. That narrow behavior makes the webhook debugging tool easy to reason about when you only need one route captured.

Key features

Tunlease focuses on a small set of routing behaviors that fit webhook debugging work:

- **Fixed callback URLs:** you keep the same callback URL instead of generating a new one every time.
- **Path-level claiming:** the command shown on the site claims a specific path such as `/demo/my-webhook/`.
- **Local forwarding:** claimed traffic is sent to `localhost:8080` in the example shown.
- **Existing endpoint preservation:** only the claimed path is redirected; other paths keep serving the original app.
- **Command-line flow:** the site shows a `tul claim` command, so the webhook debugging tool is operated from a terminal.
- **Self-hosting support:** the product includes a self-hosting guide and a GitHub link.
- **Open-source licensing:** it is marked MIT licensed.

For developers, the value of a webhook debugging tool like this is reducing setup churn. You do not need a new relay URL for every test run, and you do not have to redeploy just to observe a callback. The page also shows status text like “Connected” and “Waiting for requests,” which suggests a live relay session while the command stays running.

How to use it

Using Tunlease appears to follow a short terminal-first workflow:

1. **Start your local server** on the port you want to inspect, such as `8080` in the example.
2. **Claim a path** with the `tul claim` command, including the route you want to intercept and the relay domain shown by the app.
3. **Keep the command running** while you test incoming requests.
4. **Send webhook traffic** from your provider to the fixed callback URL and watch it arrive at your local app.
5. **Release the path** when you are done, using the terminal interrupt shown on the page.

A typical example from the site is a path like `/demo/my-webhook/` being mapped to `localhost:8080`. That means you can test webhook handlers without changing the provider's destination every time your local environment changes. As a webhook debugging tool, Tunlease is centered on that mapping step.

Who it is for

Tunlease is a good fit for people who regularly work with webhook providers and need a webhook debugging tool for local verification.

- Backend developers building payment, notification, or automation callbacks
- Engineers who need a stable endpoint during staging tests
- Open-source contributors who want a lightweight local relay setup
- Teams that prefer self-hosted tooling over a hosted SaaS tunnel
- Developers who want to keep the real application live while testing one callback route

It is especially useful when a provider requires a callback URL that should not change. If you are iterating on a handler and want to keep your local app connected to the same incoming route, a webhook debugging tool like Tunlease cuts down repetitive configuration work.

What to know before you use it

Tunlease is focused on callback routing, not on becoming a complete tunneling platform. The source material does not mention encryption details, auth features, retries, request replay, or payload inspection tools, so those capabilities should be treated as unknown unless you verify them yourself.

It is not a replacement for the official provider dashboard. If you need to confirm the provider's delivery history, signatures, or retry behavior, use the source system as the record of truth.

It routes a claimed path, not the whole site. The page explicitly says every other path keeps serving the real app, so it is intentionally narrow.

It is meant to stay running. The example warns to keep the command active while debugging, so it behaves like a live session rather than a one-click permanent setup.

Platform details are limited. The site shows macOS/Linux install instructions and a Windows option, but broader platform support is not described.

It is best for local staging, not production routing. The wording on the site points to development and staging webhook work, so production use should be evaluated carefully.

In short, Tunlease is a webhook debugging tool for developers who need a fixed callback path that reaches localhost without changing the public URL. If that is your workflow, this webhook debugging tool is designed to keep the setup simple and the callback stable.

FAQ

How much does Tunlease cost?

What does Tunlease do?

Is Tunlease free to use?

Which platforms does Tunlease support?

Is it safe to use Tunlease with production webhook data?

How is Tunlease different from tools like ngrok?

Can I host my own Tunlease relay?

Who should use Tunlease?

Sponsored

Belay – Awa... Featured
Developer Tools

Stays awake while your agents work.

Free # Developer Tools

View ↗

Partly Featured
Finance

Partly is a mobile app that lets friends quickly scan receipts, split any bill fairl...

Free # Finance

View ↗

Visit website ↗

tunlease

Pricing	Free
Category	Developer Tools
Platform	Web, macOS, Linux, Windows, API
Launched	Aug 2026
Updated	18 days ago
Verified	1 months ago

TAGS

webhook # webhooks # local-development # terminal # localhost
callback-url # debugging # cli # open source # staging # tunneling
self-hosted

Dir Hub

A curated directory of small, well-made apps. Browse by category or search to find the right one in fewer clicks.

DISCOVER

Home

Pricing

Submit your app

COMPANY

About

Contact

LEGAL

Privacy

Terms

© 2026 Dir Hub

Curated indie apps

T Tunlease